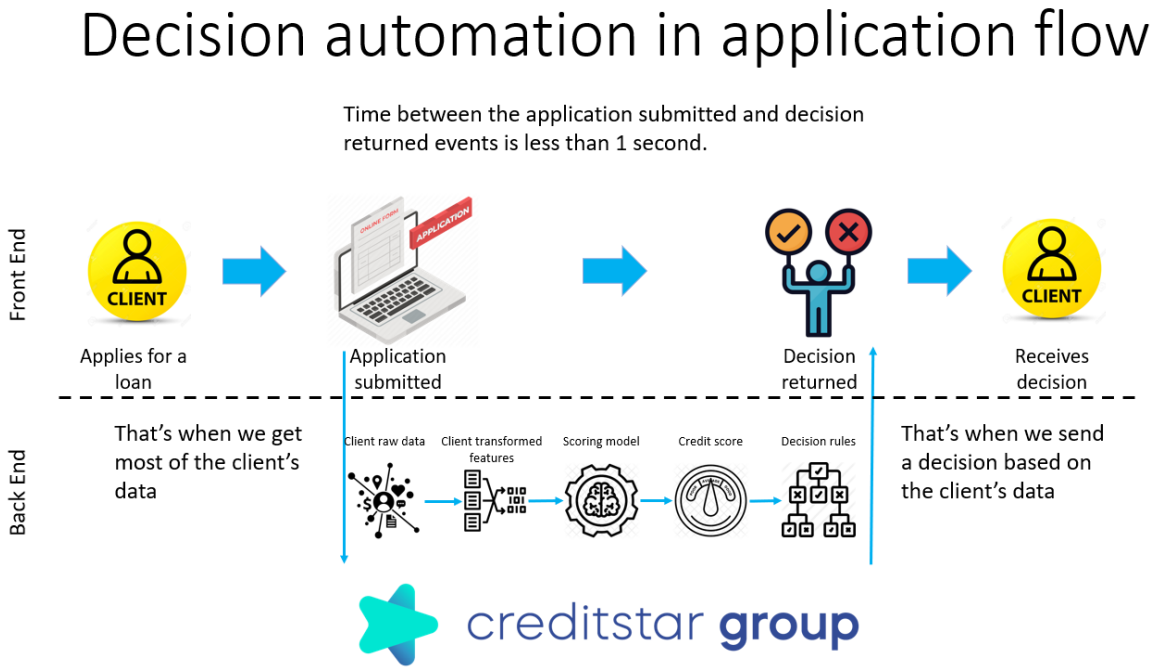
Senior Data Engineer Test Assignment

The purpose of this assignment is to evaluate your ability to both establish structure and implement elements for a data pipeline. It has been designed on purpose to reflect real world situations, where requirements can often change and the initial tasking may lack detail. Therefore, it can be enticing to spend copious amounts of time on the details. Please do not do that, don't spend more than 8 hours on this assignment. We do not expect you to submit a fully production-ready solution, however, we are excited to see your ideas and bits of implementation that show your thinking.

As a senior data engineer, one of your responsibilities will be to contribute to the development of our inhouse data services. One of them is data streaming from replicas of operational databases to S3 and making it available for other data services, such as machine learning models and decision engines. The following task reflects one of the problems that our data engineering team faces.

Business case

- We need to automate decisions made on different steps of different business processes.
- For this assignment, your task is to automate accept/reject decisions in the loan application flow.



- Client applies for a loan by submitting an online application.
- Submitted client's raw data is stored to Creditstar CRM.
- A decision automation service is called by the CRM:
    - Client's raw data is loaded to the service.
    - Client's raw data is transformed into features.
    - Features are sent to a scoring model endpoint.
    - Scoring model returns a credit score value to the service.
    - Credit score and other features are run through decision rules to make a decision.
    - Decision value is sent back to the CRM.
- CRM acts on the decision returned by a decision automation service.
- Credit decision (approve/reject) is communicated to the client.

Task (total 100 pts)
Restore the given database dump file into a Postgres database server(local) and stream the tables into a S3 bucket. If you want, you can share your S3 bucket with us, although it is not expected, as we can deduce whether and how your solution works based on the code. You will be given both plain text and binary dump files which contain the exact same data, it's up to you to decide which way to restore them, either via pgadmin or command based or any other way you prefer.

As a team we are stepping over our next hurdle of implementing a streaming service which will be used as an injection to the feature store. Most importantly, it will be the foundation of our decision engine services which will consume stored features for the decision making process. Decision services will be triggered as soon as a customer applies for a loan by submitting their data, so fetching those data from the operational database to the feature store via a streaming service must be reliable and low latency (less than 1 second).

Also, those stored features will be consumed by data scientists to train and develop machine learning models.

Models and decision engine service require derived/transformed features so the streaming pipeline should be able to handle transformations too. The transformation logic should be implemented in Python, and meant to avoid double work done by team members and reduce the time it takes to develop a feature from idea to production.

- Sub Task 1 (50 pts) :
  Implement the proposed solution technically. Technical implementation can come as an infrastructure as code (Terraform, Cloudformation) if you can. However, we do not expect you to incur any cloud platform costs when completing this assignment. Otherwise, you can submit your solution via a code repository with video guides on how it works.

- Sub Task 2 (30 pts):
  In order to make a credit decision in the business process above, data scientists requested to implement the following features in the streaming pipeline:
    - *client.paid_loans.count* - number of previously paid loans for this client.
    - *client.days_since_last_late_payment.count* - count of days since the last late payment done by this client.
    - (optional for 5 bonus points) *client.profit_in_last_90_days.rate* - sum of interest payment amount received from loans issued in the last 90 days for this client divided by sum of loan amount issued in the last 90 days for this client.

  Please use the provided data to develop the calculation logic for these features in Python and stream them to the S3 bucket using the solution developed in Sub Task 1.

- Sub Task 3 (20 pts):
  Explain the scalability of your solution. Please explain both horizontal and vertical scalability. Include a few important pros and cons of the solution.

- Sub Task 4:
  Submit the working solution. You can share your coded solution via a git repository of your preference and include a video demonstration if you would prefer.